Exercise 1F-2. Language Design [5 points].

Programming Documentation - great point that documentation should be encouraged by design. We all agree that documentation is great but at the same time we are lazy. So tools to facilitate documentations will give programmer more motivations and benefits programmers as a whole. On the native language level we have pydoc for python, that extracts function notations into reader-friendly web pages. Also I found many of the third-party tools useful. For example when I write RESTfull APIs, I Love the tool *API Blueprint* which translates a plain markdown description with little extra syntax into a nice web page. It help me keep an updated, interactive documentation of all the APIs and even help the communication between me, the backend and the frontend.

In section 3.3 the paper make a claim that *Often it encourages or even forces a programmer to split a large program into modules which are too small to express properly the structure of his problem.*. I totally disagree. From my experience, any source code file that contains more that 1000 lines of code can be split and this is also better for maintenance and future development. Now incremental compile is a widely adopted technique and source code files should be split way before it become too large for compilers.

Exercise 1F-3. Simple Operational Semantics [3 points].

$$\frac{\langle e_1, \sigma\rangle \Downarrow n_1 \quad \langle e_2, \sigma\rangle \Downarrow 0}{\langle e_1 \ / \ e_2, \sigma\rangle \Downarrow \mathit{devided\_by\_zero\_error}}$$

$$\frac{\langle e_1, \sigma\rangle \Downarrow n_1 \quad \langle e_2, \sigma\rangle \Downarrow n_2}{\langle e_1 \ / \ e_2, \sigma\rangle \Downarrow n_1/n_2 \ \mathit{if}\ n_1\ \mathit{divisible\ by}\ n_2\ \mathit{else}\ \mathit{round\_down}(n_1/n_2)}$$

Exercise 1F-4. Language Feature Design, Large Step [10 points].

$$\frac{\langle x, \sigma\rangle \Downarrow n \quad \langle c, \sigma[x := e]\rangle \Downarrow \sigma_1}{\langle \mathit{let}\ x = e\ \mathit{in}\ c, \sigma\rangle \Downarrow \sigma_1[x := n]}$$

Exercise 1F-5. Language Feature Design, Small Step [10 points].
Extended redex

$$r_{extend} ::= r \mid \mathit{let}\ x = n\ \mathit{in}\ c$$

Extra Reduction rule

$$\langle \mathit{let}\ x = n\ \mathit{in}\ c, \sigma\rangle \rightarrow \langle \mathit{temp} := x; x := n; c; x := \mathit{temp},\ \sigma\rangle$$

Extra context

$$\mathit{let}\ x = \bullet\ \mathit{in}\ c$$

1 HW1 (select all pages: your first page has your name and bookkeeping, and all others are anonymous))

- **0 pts** Correct